

# Analytical Study on Artificial Neural Network Learning Algorithms and Technique

M.Jeevitha

[jeevithamaruthachalam92@gmail.com](mailto:jeevithamaruthachalam92@gmail.com)

Assistant Professor, Department of CSE

United Institute of Technology, Coimbatore, India.

## Abstract

*The artificial neural networks are widely used to support the human decision capabilities, avoiding inconsistency in practice and errors based on lack of experience. In this paper we have explored various learning rules in artificial neural networks like perceptron learning, error correction, hebbian and competition learning rules are explored. Learning rules are algorithms which direct changes in the weights of the connections in a network. They are incorporating an error reduction procedure by employing the difference between the desired output and an actual output to change its weights during training. The learning rule is typically applied repeatedly to the same set of training inputs across a large number of epochs with error gradually reduced across epochs as the weights are fine-tuned. This paper also focuses on one of the neural network technique called Multilayer perceptron (MLP) along with its applications and advantages and drawbacks*

**Key Words:** Artificial neural networks, perceptron learning, error correction, memory based, hebbian, and competition learning, MLP

## INTRODUCTION

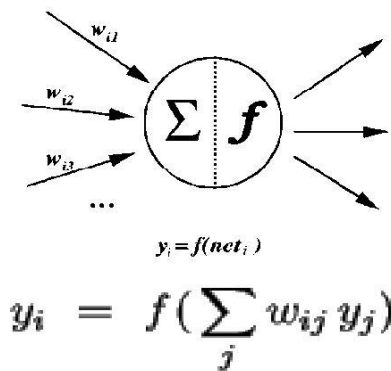
An Artificial Neural Network (ANN) is an information processing model which is stimulated by the way our brain process certain information. It is made up of

a huge amount of well interconnected processing elements namely neurons. ANNs are similar to human being in terms of learning by example. An ANN is generally configured for an application like pattern recognition or data classification which is achieved through a learning process. Learning involves repeated change or adjustments to the synaptic associations among the neurons.

The problem solving approach of Neural networks is completely way apart from the conventional algorithmic approach i.e. the computer follows a set of instructions in order to solve a problem. Computer cannot solve the problem unless and until the instructions are clear. So the problem solving capability of conventional is that we already understand and know way of solving. But it would be too good if computers could solve problems which don't precisely know how to do. Neural networks process information in way similar to what human brain does. Neural networks learn by example. It cannot be programmed to do a particular task. The drawback is that its process is unpredictable since the network finds out how to solve the problem by itself. Conventional systems are predictable; if anything goes wrong it will be because of software or hardware fault.

Neural networks and conventional algorithmic computers balance each other. Certain tasks are best suited to an algorithmic approach like arithmetic operations and there are tasks more suited to neural networks and there are number tasks which require sandwich approach (conventional+neural) to achieve with higher effectiveness. Above all it learns from the experiences collected all the way through the earlier training patterns.

The fundamental of neuron representation is often known as node or unit. Input could be from some other units, or even from an external source. Every input has its weight  $w$ , where  $w$  can be adjusted to model synaptic learning. The unit computes function  $f$  of the weighted sum of its inputs. Its output, in turn, could be input to other units. The final weighted sum of inputs is called the net input to unit  $i$ , often written  $net_i$ . Weight from unit  $j$  to unit  $i$  is denoted as  $w_{ij}$ . The function  $f$  is the unit's activation function. In the simplest case, Fig.1,  $f$  is the identity function, and the unit's output is just its net input. This is called a linear unit.



**Fig 1:** Simple Artificial Neuron Model

## 2. DIFFERENT LEARNING ALGORITHMS

### 2.1 Error-Correction Learning Algorithm

In this, the desired output value will be set and the actual output value is compared with the expected output value. The difference in turn is used to direct the training. The difference is termed as local error and the error values are used to adjust the respective weights through algorithm like back propagation algorithm. Consider the actual output as  $y$ , and the desired output as  $d$ , the error can be defined as:

$$e = d - y$$

With each training iteration the error signal is minimized to a reasonable level. In general back

propagation algorithm is used with error-correction learning.

### Gradient Descent(GD)

The gradient descent algorithm is one of the most powerful ANN learning algorithms. It is widely applied in various fields of science, engineering, and mathematics [8]. The GDA minimizes an error function  $g(y)$ , through the manipulation of a weight vector  $w$ . The cost function can be denoted as below:

Which is the linear combination of the weight vector and an input vector  $x$

$$w_{ij}[n + 1] = w_{ij}[n] + \eta g(w_{ij}[n])$$

where,  $n$  is the step-size parameter which affects the rate of convergence of the algorithm. The size of  $n$  could be too long or too small. If  $n$  is too small, the algorithm will take a long time to converge and if it is too large the algorithm might oscillate or diverge.

The GD algorithm works by taking the gradient of the weight space to find the path of steepest descent. By following the path of steepest descent at each iteration, we will either find a minimum, or the algorithm could diverge if the weight space is infinitely decreasing. When a minimum is found, there is no guarantee that it is a global minimum, however.

### 2.2 Perceptron Algorithm

Perceptron algorithm is one of the oldest and finest algorithms in Artificial Neural Networks(ANN).

It is a network in which the neuron unit calculates the linear combination of its real-valued or boolean inputs and passes it through a threshold activation function, which in case of two input variables is:

$$o = \text{Threshold}(w_0x_0 + w_1x_1 + w_2x_2),$$

$$\text{Threshold}(s) = 1 \text{ if } s > 0 \text{ and } 0 \text{ otherwise.}$$

There are chances for the systems output to vary from the desired output.

$$\text{error} = d - e$$

where, d is desired output, e is actual output.

In such cases the weights are adjusted where the activation function could be defined as follows:

$$W_{\text{new}} = W_{\text{old}} + ((\text{error}) * x_i)$$

#### Perceptron Learning Rule

1. Initialize weights at random
2. For each training pair/pattern (x, y<sub>target</sub>)
  - Compute output y
  - Compute error,  $\delta = (y_{\text{target}} - y)$
  - Use the error to update weights as follows:
 
$$\Delta w = w - w_{\text{old}} = \eta * \delta * x \quad \text{or} \quad w_{\text{new}} = w_{\text{old}} + \eta * \delta * x$$
 where  $\eta$  is called the **learning rate** or **step size** and it determines how smoothly the learning process is taking place.
3. Repeat 2 until convergence (i.e. error  $\delta$  is zero)

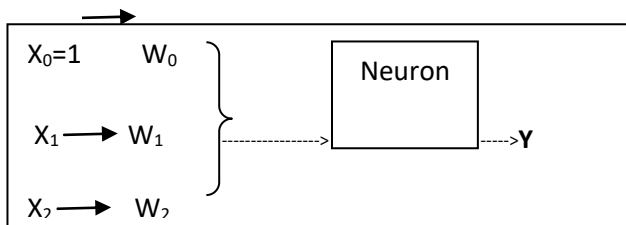
The **Perceptron Learning Rule** is then given by

$$w_{\text{new}} = w_{\text{old}} + \eta * \delta * x$$

where

$$\delta = (y_{\text{target}} - y)$$

Our single-layer Perceptron in this two-dimensional case looks like the following box where the inputs are multiplied by the weights when they enter the function.



### 2.3 Hebbian Learning

Hebbian theory is a extremely useful theory in understanding how the neurons in the brain adapt or change during the learning process. The general principle goes as follows:

The synaptic efficacy between two neurons will increase if the two neurons are 'simultaneously' active, and it will decrease if not.

According to artificial neural networks, Hebb's principle is the method of determining how to adjust the weights between model neurons. The weight between two neurons increases if the two neurons activate simultaneously, and reduces if they activate separately.

Hebb's Rule is often generalized  $\Delta w_i = \eta x_i y$ , i.e., the change in the  $i$ th synaptic weight  $w_i$  is equal to a learning rate  $\eta$  times the  $i$ th input  $x_i$  times the postsynaptic response  $y$ . The postsynaptic response can be

- Two neurons are 'positively correlated' if the presynaptic neuron fires \*before\* the postsynaptic neuron. The weight increase is greatest when this fire time difference is zero or very small. The weight increase gradually decreases as the spike time difference gets larger (but the neurons must still be 'positively correlated').
- Two neurons are 'negatively correlated' if the presynaptic neuron fires \*after\* the postsynaptic neuron, excepting for a few brief milliseconds around the 'equal correlation point'. The weight is \*decreased\* for negatively correlated spike times.
- The learning rule is the same for both the pyramidal cell and the inhibitory neuron, although this may not be true for real neurons.

### 3. MULTILAYER PERCEPTRON (MLP) NETWORK

The greater part of machine learning uses supervised learning. Supervised learning is something in which you have input variables (x) and an output variable (Y) and you use an algorithm to learn the mapping function from the input to the output.

$$Y = f(X)$$

The goal is to approximate the mapping function so well that when you have new input data (x) that you can predict the output variables (Y) for that data.

The most common neural network model is the multilayer perceptron (MLP). This is one of the best neural network algorithm for supervised network where it requires a desired output in order to learn. The goal of this type of network is to create a model that correctly maps the input to the output using historical data.

A graphical representation of an MLP is as shown below:

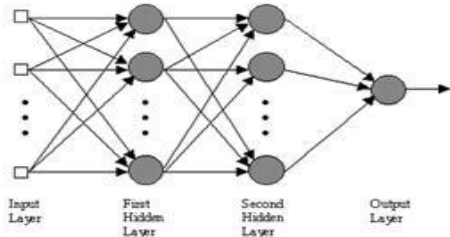


Fig: Architecture of MLP

This class of networks consists of multiple layers of computational units, usually interconnected in a feed-forward way. Each neuron in one layer has directed connections to the neurons of the successive layer. In many applications the units of these networks apply a sigmoid function as an activation function[12].

Multi-layer networks use a various learning techniques, among which the well-liked technique is back-propagation. Here, the output values are compared with the correct answer to compute the value of some predefined error-function. By various techniques, the error is then fed back through the network. Using this information, the algorithm adjusts the weights of each connection in order to reduce the value of the error function by some small amount. After repeating this process for a sufficiently large number of training cycles, the network will usually converge to some state where the error of the calculations is small. In this case, one would say that the network has *learned* a certain target function. To adjust weights properly, one applies a general method for nonlinear

optimization that is called gradient descent. For this, the derivative of the error function with respect to the network weights is calculated, and the weights are then changed such that the error decreases. Hence, back-propagation can only be applied on networks with differentiable activation functions.

## ADVANTAGES AND DISADVANTAGES OF MLP

Merits of MLP are as follows:

1. Connectionist: Can be utilized as a metaphor for biological neural networks
  2. Computationally efficient
  3. Global computing machines
  4. Adaptive learning
- The capability of learning how to do tasks based on the training data set
5. Best chosen techniques for gesture recognition.

## Drawbacks

1. Convergence could be time-consuming
2. Local minima can influence the training process
3. Difficult to scale

## APPLICATIONS OF MLP

### a) Aerospace

Use in Aerospace for:

- High performance aircraft
- Flight path simulations
- Aircraft control systems
- Autopilot enhancements

### b) Automotive

MLP is used in automotive field as follows

- Automobile automatic guidance systems
- Warranty activity analyzers.

### c) Banking

- Cheque and other document readers
- Credit application evaluators.

**d) Defense**

- Weapon steering
- Target tracking
- Facial recognition
- Image signal processing including data compression, feature extraction and noise suppression, signal/image identification.

**e) Electronics**

- Code sequence prediction
- Integrated circuit chip layout
- Process control
- Chip failure analysis
- Machine vision
- Voice synthesis
- Nonlinear modeling.

**f) Financial**

- Real estate appraisal
- Loan advisor
- Mortgage screening,
- Credit line use analysis

**g) Manufacturing**

- Manufacturing process control
- Product design and analysis
- Process and machine diagnosis
- Real-time particle identification

**h) Medical**

- Breast cancer cell analysis
- EEG and ECG analysis
- Prosthesis design
- Optimization of transplant

**i) Securities**

- Market analysis
- Automatic bond rating
- Stock trading advisory systems.

**4.CONCLUSION**

In this paper, we present the objective of artificial neural network and its applications. In this we explore the error correction, hebbian learning techniques. A detailed study on the Multilayer perceptron neural network (MLP) is presented. MLP

constructs global approximations to nonlinear input and output mapping. Consequently they are capable of generalizing in those regions of input where little or no training data is available. Each technique has its own advantages and disadvantages based on the type of neural network architecture and type of training. There is a scope for further research in this area.

**REFERENCES**

- [1] Harun, N.; Woo, W.L. ; Dlay, S.S. ;" Performance of keystroke biometrics Authentication system using Artificial Neural Net work (ANN) and Distance Classifier Method " International Conference on Computer and Communication Engineering (ICCE), 2010
- [2] Schclar, A.; Rokach, L.; Abramson, A.; Elovici Y,"User Authentication Based on Representative Users" IEEE Transactions on systems, man and cybernetics part C, Vol 42, NO. 6, pp.1669-1678,( NOVEMBER 2012)
- [3] BioPassword: History and Science of keystroke dynamics. <http://www.biopassword.com/resources/>
- [4] D.Shanmugapriya and G.Padmavathi, "An Efficient Feature Selection Technique for User Authentication using Keystroke Dynamics," IJCSNS, vol. 11, no.10, 2011.
- [5] L. K. Maisuria , C. S. Ong and W. K. Lai, " A comparison of artificial neural network and cluster analysis for typing biometrics authentication", International Joint Conference on Neural Network, IJCNN'99, vol.5, pp 3295-3299, (1999).
- [6] S. Modi and S. J. Elliott, "Keystroke dynamics verification using a spontaneously generated password password," in *Proc. 40th Annu. Int. Carnahan Conf. Security Technol.*, Lexington, KY, Oct. 2006, pp. 116–121.
- [7] A. Peacock, X. Ke, and M. Wilkerson, "Typing patterns: A key to user identification," *IEEE Security Privacy*, vol. 2, no. 5, pp. 40– 47, Sep./Oct. 2004.
- [8] N. Capuano, M.Marsella, S.Miranda and S.



- Salerno, "User Authentication with Neural networks", Univerity of Salerno Italy. [http://www.capuano.biz/Papers/EANN\\_99.pdf](http://www.capuano.biz/Papers/EANN_99.pdf)
- [9] R. O. Duda, P. E. Hart, and D. G. Stork. Pattern Classification, 2<sup>nd</sup> edition, John Wiley & Sons, Inc., 2001
- [10] S R Shorrock, D J Atkinson, S S Dlay, Biometric verification of computer users with probabilistic and cascade forward neural networks.
- [11] Ross, A., Shah, J., & Jain, A. K. March 2005. Towards reconstructing fingerprints from minutiae points. *Proceedings of SPIE Conference on Biometric Technology for Human Identification II*, 5779, 68–80.
- [12] M.S. Obaidat and D.T Macchairolo, "A multilayer neural network for computer access security", IEEE transactions on Systems, Machine and Cybernetics, vol 24(5), (1994).
- [13] S.Bleha and M. obaidat. Computer users verification using the perceptron algorithm. *IEEE Transactions on Systems, Man and Cybernetics*, 23(3):900 -902, May. 1993
- [14] D. T. Lin, "Computer-access authentication with neural network based keystroke identity verification," in *Proc. Int. Conf. Neural Networks*, vol.1, 1997, pp. 174–178.
- [15] A. K. Jain, R. Bolle, and S. Pankanti, *Biometrics: Personal Identification in Networked Society*. Norwell, MA: Kluwer, 1999.
- [16] Kohonen T. Analysis of A Simple Self-Organizing Process [M]. *Biol. cybern.*, 1982(44):135-140.
- [17] Flanagan J A. Self-organisation in Kohonen's SOM [J]. *Neural Networks*, 1996, 9(7):1185-1197.